In this issue

Computational science

With much pleasure, we bring you a special section on ‘Computational Science’ in this issue of *Current Science*. This special section was in part triggered by a local debate on the distinctions between Scientific Computing and Computational Science. The Institution of Electrical and Electronics Engineers (IEEE) uses the following definition of Computational Science to describe its magazine *Computational Science and Engineering* (now renamed *Computing in Science and Engineering*): ‘[the magazine] covers the interdisciplinary field of computational science, emphasizing the interface among applications in science and engineering, numerical and symbolic algorithms, system software and computer architecture.’ The term Computational Science presents a broader view of the activity, implying science (and engineering) that is ‘computational’ as opposed to ‘experimental’ or ‘theoretical’ in nature; Scientific Computing tends to emphasize ‘scientific’ as opposed to ‘general-purpose’ computing – a narrower perspective, although the two terms are often used interchangeably. (Computer Science, of course, deals with the science and engineering of computers.) In this special section, we interpret Computational Science in the same spirit as the above IEEE definition, and attempt to cover a good portion of the broad range of areas encompassed by the discipline.

Given the current state of the art in computing, scientists and engineers who wish to use massive computation to attack problems in their own domains cannot simply specify their massive computational needs in some high-level or symbolic manner. Typically, they have to delve into the actual engineering of the computations, and of the subsequent use of the results. This task often requires some knowledge of computer systems – computer models, computational algorithms, programming paradigms, computer architecture, computer systems software, program management tools, data management, visualization tools, etc. Two examples that illustrate this point are the lengths that mathematicians go to in optimizing their prime number detection programs on supercomputers, and the initial development of the world-wide-web and related browser software by physicists for sharing data. The typical components involved in computational science include:

- Problem formulation in computational form;
- Choice of computational technique and algorithm (for example, parallel or sequential algorithm?);
- Choice of programming paradigm and computer architecture (a plethora of choices exist for both: PVM, MPI, Fortran90, Matlab, etc.; workstations, shared-memory multiprocessors, workstation clusters, message-passing parallel computers, etc.);
- Application development, program optimization and tuning;
- Data collection and data management;
- Data analysis/visualization;
- Validation analysis, visualization of results against the real world.

In this special section, we have attempted to put together a collection of articles that cover several aspects of this broad spectrum, in the form of research papers, surveys, tutorials, and perspectives. We are indeed very happy with the response we received to the call for papers for this special section. All papers underwent a process of blind-review by typically two referees from an international set of referees, and by the guest editors. We are grateful to the referees (who remain unnamed to maintain their anonymity) for providing insightful reviews and thus helping us put together this special issue.

Overview of the articles

Problem domains as varied as studies of protein complexes, heart rate analysis, monsoon modelling and materials engineering are covered by three research papers and a survey paper. The paper by Madhusudhan and Vishveshwara (**page 852**) reports a molecular dynamics simulation of a protein complex, carried out on a 6-node SGI Power Challenge parallel computer. The simulations are used to verify stability of the protein complex and to understand protein–ligand interactions at the atomic level. The paper on ‘Computer processing of heart rate variability signals for detection of patient status in cardiac care units’ by Dutt and Krishnan (**page 864**) is an illustration of the significant inroads made in Medical Sciences through the use of simple computational tools developed in other fields. Ravi Nanjundiah’s paper (**page 869**) on ‘Seasonal simulation of the monsoon with the NCMRWF model’ gives details of a complex model that would be impossible to implement without the power of supercomputers. The survey paper by Phanikumar, Chattopadhyay and Dutta (**page 847**) on ‘Supercomputing applications in materials engineering’ provides a glimpse of how advances in high performance computing have enabled materials scientists to tackle problems of unprecedented computational complexity.

Several important *computational techniques* are addressed by papers in this special issue. ‘Numerical simulation of particulate materials using discrete element modelling’, a research paper by Sitharam (**page 876**), provides an example of a large class of problems, involving a large number of particle interactions, that are being tackled today using the discrete element modelling numerical technique – thanks to the computing power available even at the desktop. The paper on ‘A generalized algorithm for modelling phase change problems in materials processing’ by Chakraborty and Dutta (**page 887**) presents a computational framework for solving a difficult problem in materials processing that involves computational challenges on several fronts, each of which merits a full paper on its own. The tutorial paper by Arun Verma (**page 804**) on Automatic Differentiation provides enough introductory details to the reader to see why this computational method is creating waves in large-scale optimization. The tutorial article on Proper Orthogonal Decomposition by Anindya Chatterjee (**page 808**) is meant to introduce this powerful data analysis technique which has provided much hope for low-scale modelling in turbulence, one of the major remaining challenges in Computational Science. The research paper by Anbarasu, Narayanasamy and Sundararajan (**page 858**) describes a parallel genetic algorithm, implemented on C-DAC’s PARAM

10000, for the multiple molecular sequence alignment problem, an important aspect of analysing biological sequences.

Programming paradigms have a significant bearing on computational performance and ease of application development. The article by Delamarter *et al.* (**page 821**) provides an excellent tutorial on using clusters of workstations and small-scale multiprocessors, which have increasingly become a cost-effective alternative to expensive, large-scale supercomputers. The authors describe a software distributed shared memory (SDSM) system, Cashmere, that provides an easy-to-use shared-memory parallel programming abstraction on such clusters. Further, the article describes the parallelization on Cashmere of a hydrodynamics simulation program from the astrophysics domain.

Choosing a suitable *computer architecture* for the computational problem at hand is often critical for massive computations. Not infrequently, computational science problems require special-purpose computers to be built for the computation at hand, in order to realize the required magnitude of computation in a reasonable amount of time. The survey paper on ‘Reconfigurable computing’ by Bondalapati and Prasanna (**page 828**) looks at emerging reconfigurable computers whose internal architecture can be adapted from application to application, thus providing a cost-performance trade-off between special-purpose and general-purpose computers. The paper does an excellent job of covering different reconfigurable architectures as well as models of these architectures that can be used to develop suitable computational algorithms. The survey paper by Mitra and Chiueh (**page 838**) details the computational requirements of a particular area, 3D computer graphics, and describes computer architectures designed for this increasingly important application domain.

Often a computational scientist has to get involved in computational *resource management* as well. For example, load-balancing across the various processors/nodes in a parallel computer is an important factor in obtaining high computational performance on workstation and PC clusters as well as other parallel processing systems. The brief tutorial article by Qureshi and Hatanaka (**page 818**) provides a first glimpse at two load-balancing strategies in the context of a parallel raytracing application, from the image processing domain.

To round off the special section, we have an interesting perspective article by Dubey (**page 850**) entitled ‘Computing goes embedded’. The article points to a paradigm shift that will make computing ‘embedded’ in appliances and as ubiquitous as electricity. Interestingly, supercomputing class power of the past, often found in current desktops, will probably appear soon in some embedded applications. Further, many supercomputing class programming and code optimization techniques are used today in developing some embedded applications. Computational Science is already interpreted broadly enough to encompass domains such as economics and high-tech art. It will be interesting to see how broad the future coverage of computational science might be.

Sriram Vajapeyam
Rudra Pratap

Structure and tectonics of Indian Peninsular shield

Regarded as a very strong and rigid shield of great antiquity, the Peninsular India is not a monolithic landmass but an assemblage of crustal blocks multiple by variably thick belts of separated deformation, igneous activities and tectonic movements. And these blocks differ from one another in thickness, in composition, and in structure. This is now quite evident from the Deep Seismic Sounding (DSS) carried out for over three decades by the scientists of the National Geophysical Research Institute. P. R. Reddy and V. Vijaya Rao in the article on ‘Structure and tectonics of the Indian Peninsular shield – Evidences from seismic velocities’ highlight crustal inhomogeneties and point to the sites of generation and accumulation of stress in the zones of tectonic discontinuities. These sites have implication for the occurrence of earthquakes in the stable continental regions. Many of these belts are well-known from the past geological studies and quite a few have now been established beyond doubt to be of tectonic consequence and seismic potential, especially in southern and northwestern India.

For example, the wide east-west belt of rivers Kaveri, Moyar and Bhavani separating the Southern Indian Shield into two radically contrasted domains, is quite a mobile zone. And the north-south line that demarcates the boundary between the Eastern Dharwar Craton and the Western Dharwar Craton not only dismembers the Dharwar Craton into two blocks of unequal thickness and contrasted structure, but also serves as a zone of deformation dynamics. This realization, on the basis of the DSS, has a considerable bearing on the recognition of earthquake hazard-prone zones in the Southern Indian Shield. The DSS has likewise, provided a new insight into the structure and dynamics of the Aravali terrane, including the belts of strong deformation in the past and ongoing movements in the present. The next logical step in the DSS studies would be to identify and delineate precisely the zones of active tectonics and recurrent seismicity in the whole of the Indian-subcontinent.

K. S. Valdiya